MULTIMEDIA UNIVERSITY

STUDENT ID NO

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 1, 2018/2019

ECE1016 – COMPUTER & PROGRAM DESIGN

(All Sections / Groups)

13 OCTOBER 2018
9:00 A.M – 11:00 A.M
(2 hours)

INSTRUCTIONS TO STUDENT

1. This question paper consists of 12 printed pages including cover page with 4 questions only.

2. Answer ALL the questions. All questions carry equal marks and the distribution of marks for each question is given.

3. Please write all your answers in the answer booklet provided.

Question 1

(a) An example of Malaysian's income tax rate for employee in 2017 is given in Table 1(a).

Table 1(a): Income tax rate for year 2017

Chargeable Income	Calculations (RM)	Rate %	Tax(RM)
0 - 5,000	On the First 5,000	0	0
5,001 - 20,000	On the First 5,000 Next 15,000	1	0 150
20,001 - 35,000	On the First 20,000 Next 15,000	5	150 750
35,001 - 50,000	On the First 35,000 Next 15,000	10	900 1,500

Complete the C program in Figure 1(b) to prompt the user to enter the total chargeable income (RM), and then calculate the tax based on the calculation given in Table 1(a).

For example:

Ali's chargeable income = RM 24,000 (RM 20,000 + RM 4,000)
Total tax = RM 150 (for first RM 20,000) + RM 200 (5% x RM 4,000)
= RM 350.00

The program will then display the result which is the total amount tax needed to be paid in Ringgit Malaysia, RM, up to two significant digits after the decimal point. Write only the required code in the answer script. A sample output is shown in Figure 1(c).

[8 marks]

Continued...

```
// C program to calculate income tax

#include <stdio.h>

int main(){
   float income;
   float tax;

   printf("Enter total chargeable income: ");
   scanf ("%f", &income);

   // calculate the tax
   // <insert your code here>

   //display the results in 2 decimal points
   // <insert your code here>

     return 0;
}
```

Figure 1(b): Program to calculate tax.

```
Enter total chargeable income: 24000
Total income tax: RM350.00
```

Figure 1(c): Sample output. **Bold and underline** denote user's input.

(b) Write an appropriate C statement to accomplish each of the following tasks. Assume that each of these statements applies to the same program.

(i) Define a suitable structure type called `Chocolate` to store the type of chocolate sold in a superstore. Each chocolate type will have the following information:

- Name of the chocolate (up to 10 characters long)
- Unit cost price in Ringgit Malaysia
- Unit selling price in Ringgit Malaysia
- Quantity sold in number of pieces.

Use meaningful variable names and appropriate data types.

[4 marks]

(ii) Declare a structure of the type `Chocolate` called `chocList` and initialize it with the values listed in Table 1(d).

[4 marks]

Continued...

Table 1(d)

Name	Unit cost price	Unit selling price	Quantity sold
Cadbury	4.00	6.00	70
KitKat	3.50	5.00	50
Snickers	2.00	4.00	30

(c) Define a pointer, `fPtr` that points to an object of type `float`. Give two separate statements that assign the starting address of array `floaters` to the pointer variable `fPtr`.

[3 marks]

(a) Analyse the snippet of code shown in Figure 1(e) and determine the output.

[6 marks]

```
#include <stdio.h>

int main()
{
   float a = 0.002;
   float b = 0.004;
   float c, *aPtr, *bPtr;

   aPtr = &a;
   *aPtr = 3 * a;
   bPtr = &b;
   c = 4 * (*aPtr - *bPtr);

   printf ("*aPtr = %.3f",*aPtr);
   printf ("bPtr = %.3f",bPtr);
   printf ("c = %.3f",c);

   return 0;

}
```

Figure 1(e)

Continued...

Question 2

(a) The incomplete C program in Figure 2(a) allows a user to enter the size of a word to be formed from elements chosen from a character array, and then choose the characters' indexes to form the word. Complete the C program in Figure 2(a) by filling in the missing statements. The output of running the program is illustrated in Figure 2(b).

```
#include <stdio.h>
int main(){
//Question2(a)(i)
int print[5]={0};
int i, length;
//Question2(a)(ii)
printf("Choose the letters by entering their corresponding
positions:\n");
printf("r:0\n");
printf("a:1\n");
printf("l:2\n");
printf("e:3\n");
printf("t:4\n");

printf("Enter the order of characters to form the word.\n");
//Question2(a)(iii)
printf("The word you constructed is:");
//Question2(a)(iv)
printf("\n");
return 0;}
```

Figure 2(a): Program with missing statements.

```
Enter the length of the word(1-5 only): -1
Number entered must be between 1-5 only. Please reenter: 8
Number entered must be between 1-5 only. Please reenter::4

Choose the letters according to their index:
r:0
a:1
l:2
e:3
t:4

Enter the index of the characters to form the word:
Choose letter 0: 0
Choose letter 1: 3
Choose letter 2: 1
Choose letter 3: 2

The word you constructed is: real
```

Figure 2(b): Sample output. **Bold and underline** denote user's input.

Continued...

The requirements for the missing statements are specified below:

(i) Declare a character array `letters` with 5 elements and initialize the elements to `'r','a','l','e'` and `'t'`.

[2 marks]

(ii) Prompt the user to enter a value for input and read the value from the user. Validate the user input so that it must be only positive numbers between 1 to 5. If the user enters a negative number or zero, display an error message "`Number entered must be between 1-5. Please reenter:`" and obtain a new value from the user as input.

[6 marks]

(iii) Using a `for` loop to run up to the length of the word as entered by the user, prompt and scan the user input of the indexes of the letters into the array print.

Example 1: if the user enters `3` for length and wishes to create the word `'rat'`, the values in print should be `0,1,4,0,0.` (the last 2 integers are omitted as 0s because the `for` loop runs up to 3 times only).

Example 2: if the user enters `4` for length and wishes to create the word `'late'`, the values in print should be `2,1,4,3,0.` (the last integer is omitted as 0 because the `for` loop runs up to 4 times only).

Example 3: if the user enters `4` for length and wishes to create the word `'tear'`, the values in print should be `4,3,1,0,0.` (the last integer is omitted as 0 because the `for` loop runs up to 4 times only).

[3 marks]

(iv) Using a `for` loop, print out the final word using the inputs from the user as exhibited in the last line of Figure 2(b).

[3 marks]

Continued...

(b) Predict the output of the program in Figure 2(c) at the following lines:

```
1       #include <stdio.h>
2
3       int main() {
4          int num[5]={10,-5,3,-7,14};
5          printf("%d\n",*num);
6          printf("%d\n", num[4]+1);
7          printf("%d\n",*(num+1));
8          printf("%d\n",*num+3);
9          num[4]=num[1]+num[0];
10         printf("%d\n",num[4]);
11         return 0;
12      }
```

Figure 2(c)

(i) Line 5

(ii) Line 6

(iii) Line 7

(iv) Line 8

(v) Line 10

[5 marks]

(c) Assuming `str1` contains the string "`Baby`" and `str2` contains the string "`Shark`", provide the output for the following C statements:

(i)
```
printf("%s %s", str2, str1);
```

(ii)
```
if (strcmp(str2,"shark")==0)
      printf("Equivalent");
else
      printf("Not equivalent");
```

(iii)
```
printf("%c", *(str1+3));
```

(iv)
```
strcat(str1,str2);
printf("%s\n",str2);
```

[6 marks]

Continued...

Question 3

Figure 3(a) shows a complete C program. Read the program carefully, and answer the following questions.

```
1  #include <stdio.h>
2
3  void convert_fahrenheit(float);
4  void convert_kelvin(float);
5
6  int main()
7  {
8      float temperature;
9      int option;
10
11     printf("\nEnter temperature in Celsius: ");
12     scanf("%f",&temperature);
13     printf("\nConvert to:\n\t(1)Fahrenheit \n\t(2)Kelvin ")
14     scanf("%d", &option);
15
16     if (option==1)  convert_fahrenheit(temperature);
17         else if (option==2) convert_kelvin(temperature);
18         else printf("\nWrong response");
19
20 return 0;
21
22 }
```

Figure 3(a): Program to convert temperature.

Continued...

```
Enter temperature in Celsius: 16
Convert to:
      (1) Fahrenheit
      (2) Kelvin
Enter: 1
Result: 16.00 C is 60.80 Fahrenheit
```

```
Enter temperature in Celsius: 22
Convert to:
      (1) Fahrenheit
      (2) Kelvin
Enter: 2
Result: 22.00 C is 295.15 Kelvin
```

```
Enter temperature in Celsius: 12
Convert to:
      (1) Fahrenheit
      (2) Kelvin
Enter: 5
Wrong response
```

Figure 3(b): Sample output for 3 different inputs.
Bold and underline denotes user's input.

(a) For the program shown in Figure 3(a):

(i) List the function prototypes.

[4 marks]

(ii) Complete the function definitions such that the output will be similar to the sample output shown in Figure 3(b). The equations to convert the temperature T, from Celsius to Fahrenheit and to Kelvin respectively are given as follows:

$$T(°\text{F}) = T(°\text{C}) \times 1.8 + 32$$
$$T(\text{K}) = T(°\text{C}) + 273.15$$

Where °F is degrees Fahrenheit and K is Kelvin.

[14 marks]

Continued...

(b) Identify FIVE (5) errors in the code listed in Figure 3(c). List down the line number and the corrected version in your answer.
Hint: assume the line with the prototype declaration is *correct*.

```
1 #include <stdio.c>
2 void myinfo(char[], int, double, char);
3 int main(){
4 char name[20]=="Maisara Omar";
5 myinfo(name,17,'m',100108990);
6 }
7 void myinfo(char name, int age, double ID, char gender)
8 {  printf("Hello you!");
9 }
```

Figure 3(c)

[7 marks]

Question 4

(a) In C, there are some reserved keywords that *cannot be used as variable names*. Table 4(a) shows a list of such words.

Table 4(a): List of words

if	Char	integer	weight	decimal	Bob
short	Count	struct	loop	return	home
other	while	during	money	done	House
three	Monkeys	keyword	switch	for	age

Identify and list out **SIX (6) reserved keywords** that cannot be used in variable names.

[6 marks]

(b) Convert the pseudocodes shown in Figures 4(c) and 4(d) into their corresponding C program segments. As Example 4(b) shows, you do not need to write the entire program.

```
while(x < 10){
      printf("...");
}
```

Figure 4(b): Example answer

Continued...

(i) For Figure 4(c), use `if-else`.

```
If (0 ≤ price < 100)
      Print "Cheap"
Else
      If (100 ≤ price < 200)
            Print "Reasonable"
      Else
            If (200 ≤ price < 400) and (discount ≥ 50)
                  Print "Reasonable"
            Else
                  Print "Expensive"
```

Figure 4(c)

[5 marks]

(ii) For Figure 4(d), use `switch`.

```
If selection is 'S' or 's'
      Print "Saving file..."
else
      If selection is 'O' or 'o'
            Print "Opening file..."
      else
            Print "Invalid option"
```

Figure 4(d)

[4 marks]

(c) Write a complete C program that analyses the statistics of 5 social media influencers. The program should (i) prompt the user to key the number of *likes* of 5 influencers, (ii) find the minimum and maximum number of *likes* and their corresponding influencer, (iii) calculate and average number of *likes*, and then (iv) determine whether the influencer is popular or not by comparing to the average.

Continued...

To achieve the FOUR (4) objectives, you can use the following suggestions (not compulsory).

(i) Using a loop, the program first prompts the user to key in the number of *likes* for 5 influencers and stores them in an array. Assume that the user will always key in more than 0, therefore the program does not need to check the input;

(ii) Assign one of the 5 number of *likes* as the minimum and then compare it with the others. If smaller, replace the minimum number of *likes* and its corresponding influencer.

(iii) Assign one of the 5 number of *likes* as the maximum and then compare it with the others. If bigger, replace the maximum number of *likes* and its corresponding influencer

(iv) Sum up all the *likes* in the array and divide by 5 to obtain the average;

(v) Finally, using a loop, the program compares the average *likes* to the *likes* in the array. If the number of *likes* is above or equal to the average, print POPULAR, else print NOT POPULAR.

A sample of the output is shown in Figure 4(e).

Hint: You have to use an array to write this program.

```
Enter no. of likes for Influencer 1: 540
Enter no. of likes for Influencer 2: 760
Enter no. of likes for Influencer 3: 950
Enter no. of likes for Influencer 4: 920
Enter no. of likes for Influencer 5: 120
Minimum: Influencer #5 (120.00 likes)
Maximum: Influencer #3 (950.00 likes)
Average likes: 658.00
Student 1: NOT POPULAR
Student 2: POPULAR
Student 3: POPULAR
Student 4: POPULAR
Student 5: NOT POPULAR
```

Figure 4(e): **Bold and underline** denotes user's input

[10 marks]

End of Paper